TECHNICAL REPORT 1976
December 2008

Rapid, Agile Modeling Support for Human–Computer Interface Conceptual Design

J. DiVita
R. Morris
G. Osga

SSC Pacific

TECHNICAL REPORT 1976
December 2008

Rapid, Agile Modeling Support for Human–Computer Interface Conceptual Design

J. DiVita
R. Morris
G. Osga

SPAWAR
Systems Center
PACIFIC

SSC Pacific
San Diego, CA 92152-5001

SSC PACIFIC
San Diego, California 92152-5001

M. T. Kohlheim, CAPT, USN
Commanding Officer

C. A. Keeney
Technical Director

ADMINISTRATIVE INFORMATION

This report was prepared for Office of Naval Research (Code 31) and the SPAWAR Systems Center Pacific (SSC Pacific) Science and Technology Initiative by the Command & Control Technology & Experimentation Division (Code 536).

Released by
T. Tiernan, Head
Command & Control Technology
& Experimentation Division

Under authority of
T. Tiernan, Head
Command & Control Technology
& Experimentation Division

SB

EXECUTIVE SUMMARY

SPAWAR Systems Center Pacific (SSC Pacific) has developed prototype combat systems. The method of evaluating design alternatives for these prototypes is through an iterative cycle of design, build, and test. This approach has several drawbacks. For example, design information that is critical to support cognitive and perceptual processes in the task domain is not explicitly captured by the design process, but rather is implicitly embodied in the final design. To overcome the problems of this approach, SSC Pacific has explored the feasibility of a model-based approach to system design.

A model-based approach to interface design requires a cognitive architecture. A cognitive architecture is a "fixed set of components and mechanisms that represent basic human abilities and limitations" (Kieras, 2005). Because cognitive architectures provide a set of performance constraints, they may be used to predict human performance for various tasks. These architectures have been incorporated into engineering models that predict usability testing of interface design.

At SSC Pacific, we have used the engineering model GOMS to evaluate interface design. The acronym "GOMS" (Card, Moran, and Newell, 1983) stands for Goals, Operators, Methods and Selection rules. A GOMS keystroke-level model represents the series of keystrokes, visual searches, cursor moves, and mouse-clicks the operator must perform in order to accomplish a task with the interface. A keystroke-level evaluation presumes an explicit design and simulates an operator performing a particular task using the interface. The model requires a strategy to perform the task. From these data, reaction times to perform sequences of operations are added to compute times required for specific tasks. These reaction time data may serve as a way to evaluate the system. The use of this model has led to the following lessons learned: (1) Modeling tools focus their analysis on mature designs and do not guide early conceptual design for positive human factors engineering impact. (2) Modeling tools are often slow and cumbersome for rapid iterative design cycles. It takes too long to capture the Human–Computer Interaction. (3) Keystroke-level models do not map well to usability testing in rapid-prototyping methods. For example, a "walk-up test" is a usability test used to evaluate the intuitiveness of a design. The operator is given a task to perform on a new and unfamiliar interface. A keystroke model cannot perform this very valuable type of usability testing because this type of model must be programmed with a particular strategy to perform a task. Since these models cannot "look" at an interface and "think" of a strategy to perform a task, they cannot begin to predict intuitiveness or the ease of use on an interface, but the walk-up usability test is very informative and a powerful aid to the early design process.

Therefore, what is needed is a cognitive architecture that matches, in time and scope, the iteratively cyclical design process and is suited to predict human performance in early design usability testing. Since this performance entails aspects of higher level perceptual and cognitive processing, these architectures must constrain perceptual and cognitive performance along the lines of higher level processing necessary to evaluate interface design. This report reviews several models that may be suited to predict early design usability testing, of which the walk-up test is an example. One model, Automated Cognitive Walk-through for the Web (ACWW) is used in a preliminary evaluation of the Knowledge Web (KWeb) Interface.

CONTENTS

Figures

Tables

1. BACKGROUND

SPAWAR Systems Center Pacific (SSC Pacific) has developed prototype systems in Air Defense Warfare (MMWS), Land Attack Combat Systems (LACS), and Intelligence Analysis (KRSOC). The method of evaluating design alternatives for these prototypes is through usability testing. An iterative cycle (design, build, test) is currently employed. This approach has several drawbacks. For example, a precise method to prescribe a display layout or design based on the task information requirements does not exist; thus, the iterative testing of hypothesized best layouts is required. The “size” of the design space and the constraints of the design space are unclear and unbounded. Another problem is that the proof of the value of these design hypotheses lies solely in usability testing and data collection. The degree to which this testing can be effectively done is debatable since time constraints pose various limitations. For example, a small number of test subjects and prototypes with limited fidelity are typical drawbacks for these studies. It is hoped that design alternatives will evolve to contain information that is considered critical to support cognitive and perceptual processes for each task domain. Unfortunately, this information is not explicitly captured by the design process, but rather is implicitly embodied in the final design.

At best, this design process can produce a heuristic set of “lessons learned” and a usable interface that meets task performance requirements. As viewed from the most negative perspective, this design process may require many cycles of empirical testing of ad hoc systems that is terminated when project resources are expended or when performance results are finally achieved. Unfortunately, if resources are expended, a design that is just “good enough” may be accepted versus one that is optimal for task conditions. To overcome the problems and pitfalls of this approach, SSC Pacific has explored the feasibility and usability of a model-based approach to system design.

2. A MODEL-BASED APPROACH TO INTERFACE DESIGN AND USABILITY TESTING

A cognitive architecture is a “fixed set of components and mechanisms that represent basic human abilities and limitations” (Kieras, 2005). Cognitive architectures may be composed of perceptual, motor, and cognitive processors. For example, perceptual processors model visual and auditory capabilities. Motor processors model ocular, vocal, and manual motor capabilities. Cognitive processors model long-term memory, working memory, production rule memory, and problem solving capabilities. Because cognitive architectures provide a set of performance constraints, they may be used to predict human performance for various tasks that require perceptual, motor, and cognitive activity. Thus, tasks typically studied in a psychology laboratory may be modeled and compared to actual human performance data. Various aspects of these architectures have been incorporated into engineering models that predict usability testing of interface design. The purpose of a model-based evaluation of an interface is to save time and money by predicting operator performance before an actual prototype is built.

For example, GOMS is an engineering model for interface design that attempts to explicitly represent “procedural knowledge”—that is, the knowledge a user must have to perform certain tasks. The acronym GOMS (Card, Morgan, and Newell, 1983) stands for goals, operators, methods, and selection rules. Thus, the operator may have a goal to accomplish a task. The operator uses certain methods to accomplish this goal. The methods utilize basic operators in a series of steps that the user performs. The appropriate method is chosen by selection rules that reflect the current operating context.

Various levels of GOMS models exist (for example, high-level GOMS models, Cognitive Perceptual Motor (CPM)-GOMS, keystroke-level GOMS). At the lowest level, the GOMS keystroke-level model represents the series of keystrokes, visual searches, cursor moves, and mouse-clicks the operator must perform to accomplish a task with the interface. A keystroke-level evaluation presumes an explicit design and simulates an operator performing a particular task using the interface. The model requires a strategy to perform the task. Deriving an explicit strategy to perform a task is often quite revealing of the usability of an interface. Data are collected at the keystroke level of operator (model) performance. From these data, reaction times to perform sequences of operations are added to compute times required for specific tasks. This reaction time data may serve as a way to evaluate the system.

Kieras (1997) developed NGOMSL, which takes the procedural knowledge captured in a keystroke-level GOMS model and represents that knowledge in the form of production rules used in cognitive architectures. Santoro and Kieras (in preparation) modeled a team of four operators performing an Air Defense Warfare (ADW) task with a team of GOMSL models. At SSC Pacific, we have built three different ADW teams and a Land Attack Operator with GOMSL models. This process has lead us to the following set of lessons learned.

2.1 LESSONS LEARNED

2.1.1 Modeling Tools' Focus on Analysis of Mature Designs Cannot Guide Early Conceptual Design for Positive Human Factors Engineering Impact

A keystroke evaluation is ideal for human–computer interface (HCI) tasks because these tasks rely heavily on perceptual motor activity. The question is, what is the relationship between this type of analysis and the HCI design process? One drawback is that a keystroke-level of analysis occurs at the very end of the design process. The interface has been specified; thus, using this method as an approach to design is a bottom-up approach. At SSC Pacific, several projects, including Pacific Command Hawaii (PACOM), Joint Intelligence Command Pacific (JICPAC), Hawaii Security Operations Center (HSOC), and White House Situation Room (WHSR), have adopted the top-down approach of "Usage Centered Design" (Constantine and Lockwood, 1999). In usage-centered design, the layout of the actual interface is discouraged in the early design phase. Instead, the designer is engaged in a rather abstract system specification, creating an "abstract prototype" (Constantine, 1998). For example, a list of user intentions and system responsibilities may first be mapped out, which means that modeling techniques that attempt to replace usability testing at the keystroke level of evaluation are inadequate to guide early conceptual design of interfaces.

2.1.2 Keystroke-Level Models Do Not Map Well to Usability Testing in Rapid-Prototyping Methods (Not a Reaction-Time Problem)

One aspect of usability testing that is popular in industry and used at SSC Pacific is to test the intuitiveness of an interface, which is sometimes referred to as a "walk-up test." The idea here is, how much could the operator accomplish if he/she just walked-up to the interface and was given a task to do with a minimal amount of instruction? A keystroke model cannot perform this very valuable type of usability testing because this model must be programmed with a particular strategy to perform a task. Since these models cannot "look" at an interface and "think" of a strategy to perform a task, they cannot begin to predict intuitiveness or the ease of use on an interface, but the walk-up usability test is very informative and a powerful aid to the design process.

2.1.3 More Global Metrics and Methods are Needed for Usability Test Performance Predictions (Keystroke Models are Scaled Wrong)

Usability testing is often used to corroborate a heuristic evaluation method (Nielsen and Molich, 1990 and Nielsen, 1993). For example, Nielsen lists the following usability heuristics:

1. Use simple and natural language
2. Speak the user's language
3. Minimize user memory load
4. Be consistent
5. Provide feedback
6. Provide clearly marked exits
7. Provide shortcuts
8. Provide good error messages
9. Prevent errors
10. Use gestalt principles for graphic design and color. (This heuristic was not specifically listed with those above but is discussed extensively in Nielsen (1993) in the context of this list.)

A glance at usability heuristics reveals a list of interface features that clearly deal with higher order perceptual and cognitive processing. Reaction time plays a minor role, if any, in such an evaluation. Obviously, consequences pertaining to these heuristics will be reflected in reaction time, but the point is, the usability tester is not measuring reaction time, but rather conducting the evaluation at a higher level of operator performance. Thus, screen layouts are evaluated to see if they follow rules of gestalt grouping. For example, are display elements that "belong" together in terms of functionality, close enough to one another, spatially and temporally, to form a visual group? Is there a good "mapping" between the user's "mental model" and the visual presentation of information? Is the interface consistent? Is the interface awkward?

Interestingly, several of these heuristic evaluations could be performed, and their outcome expressed in a quantitative manner by the keystroke-level models; however, the evaluations are currently "buried" in the reaction time data and must be realized by the designer examining the data and seeing the appropriate relationships. In summary, the output of keystroke-level models is currently not at the level of a simple design principle that the designer can readily use. The models do not make the higher level connection between data and design principle that is necessary to ensure good design practices.

2.1.4 Current Usability Level of the Models

The modeling we have done is too slow and cumbersome for rapid iterative design cycles. It takes too long to capture the HCI. The HCI must be mature enough to be captured (refer back to Section 2.1.1), making change limited to only egregious errors discovered by the modeling. It is also too time consuming to make the design-test scenario run. Usability testing of human participants is actually faster than keystroke-model usability testing. To conclude, an engineering model that matches, in time and scope, the iteratively cyclical design process is needed.

3. CHALLENGES

3.1 BUILD AN ENGINEERING MODEL THAT MATCHES, IN TIME AND SCOPE, THE ITERATIVELY CYCLICAL DESIGN PROCESS, WHICH IS A TIMED TEST

Two actual prototypes presented in Figures 1 and 2 show how daunting the challenge is to keep pace with an actual design process. These two prototypes represent the extremes of the design space in that four other design alternatives exist that span a continuum of functionality and design between interfaces 1 and 2. That said, there are more than six alternatives. The designers stopped at six because they “felt” these were different enough from one another to be considered as “alternatives.”

Interface 1 (Figure 1) consists of a spreadsheet data field (top window) and two workspaces below. A row in the top data field is selected and changes are made to the cells of the row in the left workspace below. Notes and comments may be added in the right workspace below. The initial criticism of this design was that the editing should be more direct. The up and down eye and mouse movements from data spreadsheet to workspace should be avoided. Figure 2 represents a design where all of the editing to the spreadsheet may be accomplished by pull-down menus attached directly to the cells of the worksheet. The pull-down menus alleviate memory requirements for the permissible cell values. Initial reaction to this interface was that it was “ugly”—too cluttered. (Note the comparison between the degree of clutter in Figures 1 and 2 that the designer is making is not at all fair in that the designer has conveniently chosen to present considerably more rows of data in Figure 2 as opposed to Figure 1.)

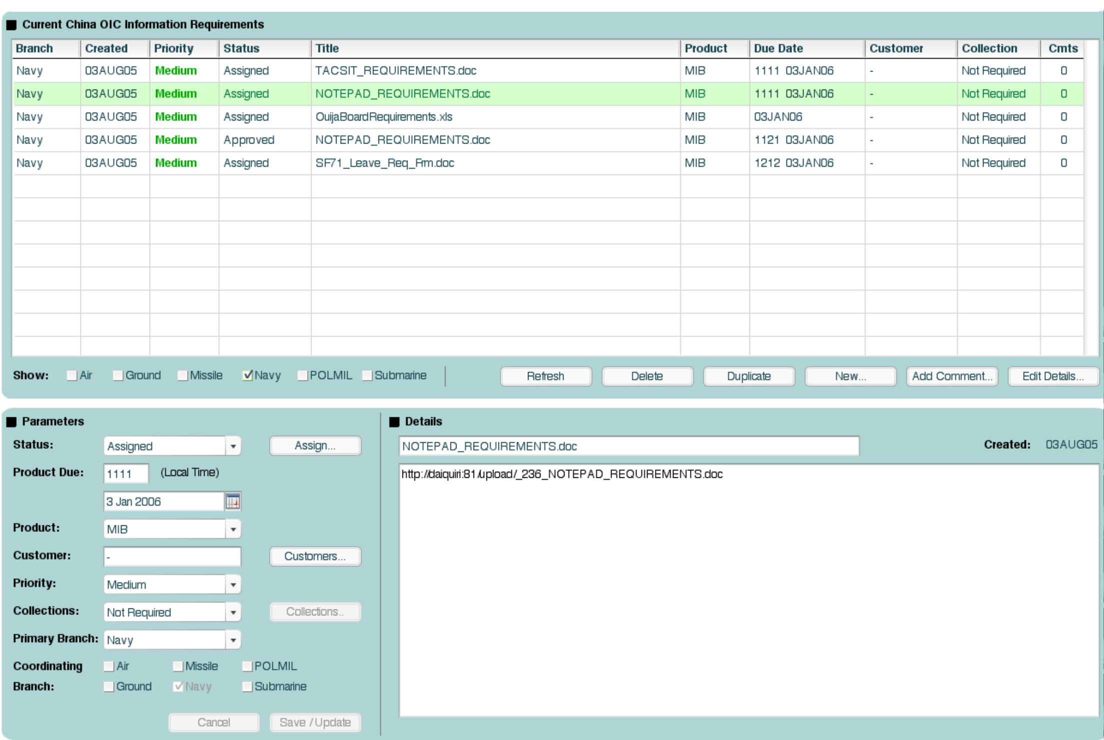
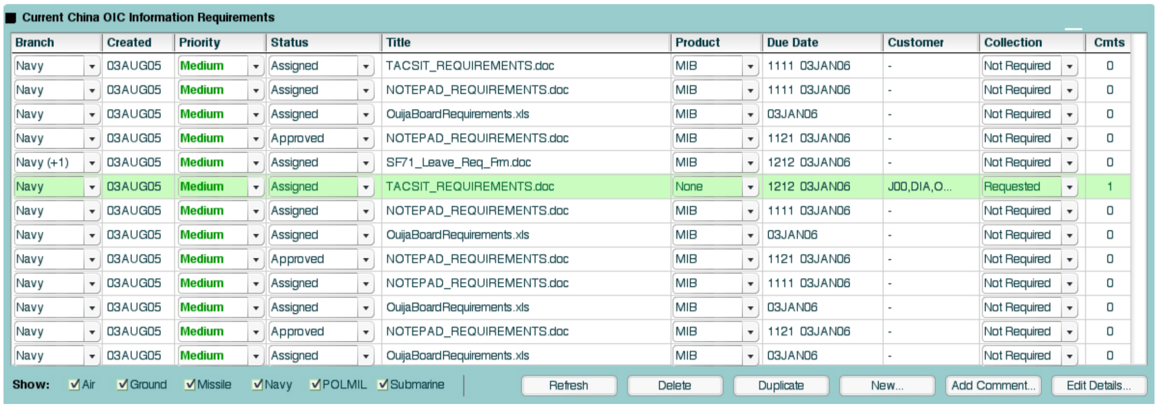


Figure 1. Paper prototype of an interface.



Current China OIC Information Requirements

Branch	Created	Priority	Status	Title	Product	Due Date	Customer	Collection	Cmts
Navy	03AUG05	Medium	Assigned	TACSIT_REQUIREMENTS.doc	MIB	1111 03JAN06	-	Not Required	0
Navy	03AUG05	Medium	Assigned	NOTEPAD_REQUIREMENTS.doc	MIB	1111 03JAN06	-	Not Required	0
Navy	03AUG05	Medium	Assigned	OuijaBoardRequirements.xls	MIB	03JAN06	-	Not Required	0
Navy	03AUG05	Medium	Approved	NOTEPAD_REQUIREMENTS.doc	MIB	1121 03JAN06	-	Not Required	0
Navy (+1)	03AUG05	Medium	Assigned	SF71_Leave_Req_Frm.doc	MIB	1212 03JAN06	-	Not Required	0
Navy	03AUG05	Medium	Assigned	TACSIT_REQUIREMENTS.doc	None	1212 03JAN06	J00,DIA,O...	Requested	1
Navy	03AUG05	Medium	Assigned	NOTEPAD_REQUIREMENTS.doc	MIB	1111 03JAN06	-	Not Required	0
Navy	03AUG05	Medium	Assigned	OuijaBoardRequirements.xls	MIB	03JAN06	-	Not Required	0
Navy	03AUG05	Medium	Approved	NOTEPAD_REQUIREMENTS.doc	MIB	1121 03JAN06	-	Not Required	0
Navy	03AUG05	Medium	Assigned	NOTEPAD_REQUIREMENTS.doc	MIB	1111 03JAN06	-	Not Required	0
Navy	03AUG05	Medium	Assigned	OuijaBoardRequirements.xls	MIB	03JAN06	-	Not Required	0
Navy	03AUG05	Medium	Approved	NOTEPAD_REQUIREMENTS.doc	MIB	1121 03JAN06	-	Not Required	0
Navy	03AUG05	Medium	Assigned	OuijaBoardRequirements.xls	MIB	03JAN06	-	Not Required	0

Show: Air Ground Missile Navy POLMIL Submarine | Refresh | Delete | Duplicate | New... | Add Comment... | Edit Details...

Figure 2. Paper prototype of alternative design of interface depicted in Figure 1.

The next step in the design process would be to conduct usability testing (hopefully a fair test) on a subset of the six alternatives. Two questions must be asked about this testing: (1) could this testing be performed with a model in a timely enough fashion to impact the choice of the interface? (2) can the model provide data and feedback that address the concerns of the designers? For example, how would the model address the claim that interface 2 is too cluttered? One simple and direct way would be to compare visual search times in the two interfaces, but that would require a model of visual search. GOMSL, for example, simply attributes a constant visual search time (1.2 seconds) to all visual searches that are preformed in a procedure. So the underlying cognitive architecture of GOMSL is not robust enough for this comparison. Perhaps a model that determines perceived "numerousity" would be appropriate to measure clutter. Again, the problem is whether the model could perform such an analysis. Thus, we are led to problem 2 in Section 3.2.

3.2 ARE THE COGNITIVE ARCHITECTURES UP FOR THE CHALLENGE?

Usability testing is often concerned with higher level perceptual and cognitive aspects of the interface. For example, modeling visual search or perceptual grouping is a daunting task. Theories on either topic may make claims extending from the earliest stages of visual processing to the phenomenology of consciousness. Visual search has been the focus of intense investigation for the past 25 years, and perceptual grouping for nearly a century. Theories and controversy abound for each topic at every stage of processing, and theories exist that would reject the notion of stages of processing altogether. The challenge is to find a level of reliable experimental data that could be incorporated into a perceptual architecture adequate to guide interface analysis. To conclude, building the necessary architectures that will constrain perceptual and cognitive performance along the lines of higher level processing necessary to evaluate interface design is a challenge.

3.3 DATA FROM AN ENGINEERING MODEL DO NOT EQUAL GOOD DESIGN PRINCIPLES

Making the connection between the quantitative data that an engineering model can produce and good design principles is an ongoing experimental and research endeavor. To bridge model data to design principles is a challenge.

3.4 DESIGNING INTERFACES FOR TEAMS OF OPERATORS

Lastly, several researchers (Alterman, 1999; Kieras, 2005) have pointed out that most interface modeling endeavors are concerned only with an individual and a computer; however, individuals typically work and collaborate with others in some form of a team structure. Little work has been done to capture interface designs that explicitly aid the collaborative nature of work. Models that capture the social aspect of work so that it may be reflected in interface design poses yet another challenge.

4. APPROACH

Figure 3 is a rough sketch of a timeline for the development of a new interface. The process starts with the designers specifying the functionality of the interface—what will it be used for and who the users will be? The users tasks and task sequences are specified early in the process. These initial phases are “pre-interface,” that is, early in the design process, no interface exists. Eventually, paper prototypes of the interface are constructed. After some form of usability testing, these paper prototypes will serve as the basis for early interface designs with limited functionality. Eventually, the tasks and the interface are specifically defined and a fully operational interface is produced. For each step of the process, we may ask, “what tools are available to help the interface designer? What tools are available to ensure that good design practices are being implement at every stage of development?”

Our past research has focused on keystroke-level design analysis tools. These tools are suited for analyzing a mature interface design for a completely specified task. The goal of this research will be to develop tools that are suited for earlier stages of the design process. In particular, we will focus on developing tools that will analyze early paper prototype designs. This focus will require increased emphasis on higher level cognitive processes as they relate to interface design. In addition, our modeling efforts will be mapped to a rapid, iterative “agile design” process.

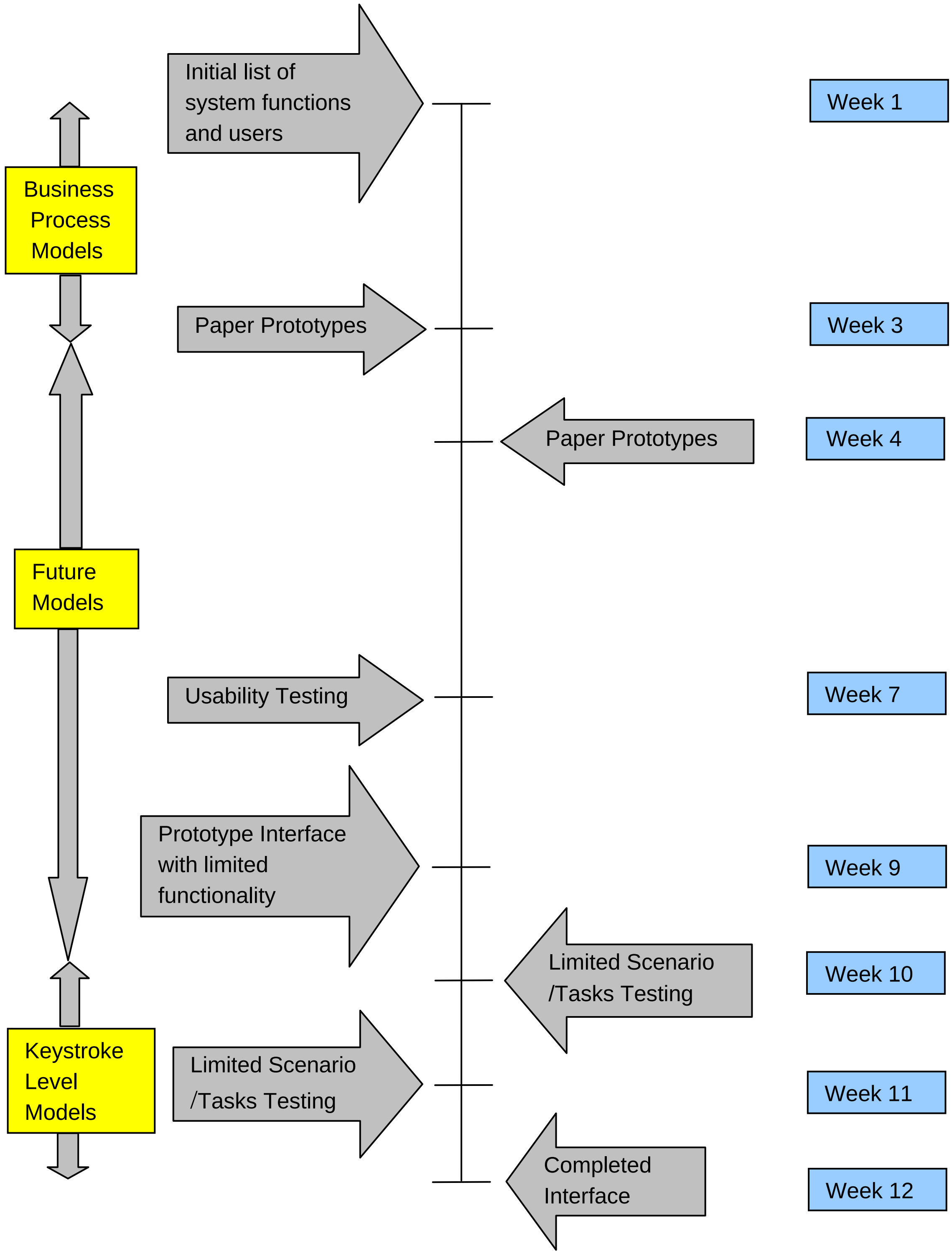


Figure 3. Timeline of a 3-month interface development cycle.

4.1 THE AGILE DESIGN PROCESS

The "agile design" process is based on rapid and X-treme programming methods and usage-centered design. The agile design process has been used in most recent projects at the SSC Pacific User-Centered Design (UCD) Group. Our work will focus on design tools that may be used to facilitate the agile design process.

The software development process referred to as "agile" has seen increasing use in recent years as an attempt at finding increased efficiency from concept to market in software product development. Fowler (2005) describes the process as "adaptive" rather than "predictive" and the methods as "people-oriented" rather than "process-oriented."

First, the predictive nature of the process separates the design and construction of the software and places emphasis on planning (designing) before building. The software process is often creative and unpredictable as requirements are uncovered during usability testing early in the process using paper prototypes. Thus, planning must be agile and adaptive to changing requirements.

The second aspect noted by Fowler refers to the design team members' interaction. Notably, the process involves a programmer, an HCI designer, a human factors engineer and an end-user in the same early stages of the design process. Given the dynamics of design and lack of predictability, iterations become necessary, and the team works iteratively through the problems as a unified entity. The length of iteration varies with different agile approaches—anywhere from 1 month to 6 months, for example. This iteration has important implications, specifically, that modeling tools and approaches that cannot quickly respond to design trade-off questions are not as useful in providing design support, or provide too little, too late.

The agile design process includes the following steps:

- Mission Statement—statement of purpose of the system to be built
- Domain Model—synopsis of the work environment, users, locations, customers
- Role Model—description of users' roles in the system domain
- Essential Stories—short narrative that provides a storyboard of system use
- Task Descriptions—descriptions of tasks to be accomplished and task goals, technology independent
- Activity Diagrams—sequential flow diagram of user and system activity
- Design Test Case—metrics or observations that will be used to test the design
- Content Model—information content to express requirements at each task step
- Wire Frame/Canonical Prototype—general HCI format and design to place content
- Prototype HCI Design—focus on content, layout, and functionality without focusing on widget design
- Usability Evaluation Plan—storyboard and question or prompt list for testing purposes
- Prototype—iterations of working HCI software as system is built
- Task Flow Diagram—walkthrough of the HCI in use by user
- Usability Evaluation & Report—results of testing at each development spiral
- Incremental Feature Map—time-related mapping of features to each spiral

Design tools that can predict cognitive or visual performance are most useful during early HCI spirals where critical design trade-off decisions are made. At this stage in the design process, key features of the HCI are selected for the system and then subjected to usability testing. The decisions the designer makes in selecting and rejecting HCI features can result in erroneous rejection of useful and efficient design alternatives. Tools would be highly beneficial that would allow efficient comparison of alternative design approaches at these early stages of concept formation. The tools must be able to receive input related to format and design content and storyboard content (use). The tools should also contain an embedded model of human performance.

4.2 PRINCIPAL METAPHORS AND ASSUMPTIONS OF OUR MODEL

Testing paper prototypes typically entails the usability walk-up test described Section 2.1.2. In this usability test, the participant is given a task to perform on a novel interface and must generate a user's strategy (a series of interactions with the interface) to accomplish the given task. The "interface" is sketched on paper and the user "selects" items by pointing to objects. We hypothesize that modeling usability at this level entails describing a user's search process that allows the user to explore the interface. This search process entails mapping user intentions—what task the user intends to perform with the system—onto a set of "affordances" that the interface must exhibit in order for operators to act on their intentions. "Affordances" was a word coined by Gibson (1979) to specify what the environment "affords" the organism—that is, what the environment will allow the organism to do. In this context, interface affordances refer to those features of the interface that suggest to the operator how to use the system. Effective interface design requires a "good" mapping of interface affordances to user intentions.

As the system design matures from an abstract representation of affordances to particular interface objects and "widgets" that instantiate system affordances, a saliency map may be constructed. Saliency maps have been developed to model guided visual search (Wolfe, Cave, and Franzel, 1989). These models associate a level of activation to each object in the visual field. The activation level of an object is a function of the goals (intentions) of the observer's visual search. For example, if the observer is searching for a red square, then "redd-ish" and "square-ish" items receive a greater level of activation than items with a different color or shape. The activation levels of objects in the visual field produce a saliency map that ranks what objects should be attended to first in order for the observer to find the target. By comparing the level of activation of the target item to that of the distractor items, a signal-to-noise ratio is generated that predicts search efficiency. Similarly, the relative efficiency with which an interface guides an operator to achieve their goal may be modeled and evaluated. Given a strategy—a specific set of user interactions with the interface—the saliency of interface affordances needed to carry out this strategy would be evaluated by the model. Various models concerned with interface comprehensibility and user exploratory behavior have attempted to provide such saliency metrics. We will examine the usefulness and predictive character of these models.

5. SURVEY OF RELEVANT MODELS

5.1 CONSTRUCTION-INTEGRATION THEORY OF TEXT COMPREHENSION

Kintsch (1988) has argued that text comprehension is a cyclical process. A reader comprehends a sentence, or a sentence fragment, in one construction-integration cycle. Comprehension entails a sequence of these cycles. On each cycle, Kintsch's model considers (1) the reader's goals, (2) the elements comprehended thus far, and (3) a propositional representation of the text to be comprehended.

The comprehension cycle is a two-phase cycle. The first phase consists of a network of propositions that contain the possible meanings of the current sentence. A "construction" process generates the network. The nodes of this network are representations of the input words from the text, the various meanings of these words that have been retrieved from long-term memory (LTM), the reader's goals, and the current context. This process is considered "bottom-up." The second phase of the cycle integrates the information assembled in the first stage. A meaning for the current sentence is determined by a "spreading activation mechanism" that considers the current context. Thus, the most highly activated nodes, computed from a combination of the reader's goals and the current context, represents the text's meaning.

5.2 NETWORK

The construction-integration theory of text comprehension was extended by Mannes and Kintsch (1991) to action planning in their NETWORK model. They hypothesized that action planning and text comprehension are similar in the sense that a meaning or an action must be selected from a set of competing meanings or actions. They applied their model to the domain of user–computer interaction.

5.3 LICAI MODEL

To model a user's exploration of a novel interface, Kitajima and Polson (1995a, 1997) originally proposed the "LInked model of Comprehension-based Action planning and Instruction-taking" (LICAI). Kitajima and Polson's model follows Hutchins, Holland, and Norman's (1986) interpretation of direct manipulation (Shneiderman, 1982). Direct manipulation in this context refers to the interactions between a "user" and a graphical computer interface. For Hutchins, Holland, and Norman, direct manipulation is an action cycle.

5.3.1 A Complete Action Cycle

Kitajima and Polson describe a "complete action cycle" for their model, which consists of four components: (1) goals (a sequence of subtasks or nested goals that will allow the user to accomplish his or her overall goal), (2) the world, (3) the stage of evaluation (the user evaluates the display before taking an action), and (4) the stage of execution, which includes the processes that are necessary for the user to actually take an action. Once an action is taken, the interface, "the world," will change.

5.3.1.1 Goals

Goals are divided into two types: (1) task goals, and (2) device goals. Associated with each of these goals are states; thus, there are tasks states and device states. The user must search both state-spaces and map one state-space to the other. For example, the device-state is the state that the interface must be in so that the user may accomplish a specific task and arrive at a specific task-state. In this manner, the two states are "yoked" (see Payne, Squibb, and Howes, 1991).

For users to successfully accomplish a task on an interface, they must have the correct task and device goals. The latter is a key to good interface design, that is, a new user does not know, a priori, the necessary device goals of an interface. The interface must lead the user to easily explore and determine what the device goals are for a given task. In the initial application of Kitajima and Polson's model, the model was given the correct device goals. These goals were placed in LTM and represented the knowledge of the interface acquired by the experienced user. In their later LICAC model, LICAC+, the device goals were eliminated and the model had to generate device goals (see Section 5.4 and 5.5).

5.3.1.2 The World

In this case, the world is the interface. The interface will react to the user's actions by modifying its display.

5.3.1.3 Stage of Evaluation

Two processes exist that underlie the stage of evaluation: (1) generation of the display representation—analogous to Kintch's construction phase, and (2) elaboration of the display representation—analogous to integration. This elaboration leads to the model's evaluation of information presented on the display.

a. Generation of the display representation. The display is parsed into objects and the spatial position of these objects is known. Some perceptual attributes of each object are also known by the model. What the model does not know is the relationships between objects or the functions of objects on the display.

b. Elaboration of the display representation. The user must form links, relationships between their task goals and the display objects. At first, the model does not assume the user knows any of these relationships. Users must link the task to items that are currently displayed on the screen and actions to be taken on those items so that their task moves closer to being accomplished. Building these links simulates the user's evaluation of the display. The model builds these link through a memory sampling process that is analogous to the spread of action mechanism in Kintch's model. Objects on the display act as retrieval cues into the model's LTM. Stored in the model's LTM are the representations of goals needed to perform various tasks. The retrieved information provides information about the display such as interrelationships about display objects and relationships between tasks and display objects.

An evaluation of the display follows every action. This evaluation must consider the task goals and the device goals. The elaboration process is probabilistic. The model can make an error if the elaboration process does not provide the correct information, that is, the model may fail to remember key relationships between the interface and the task goals.

5.3.1.4 Stage of Execution

The model assumes two processes. The first process is the selection of candidate objects (or actions). The model then generates three possible objects on which to act. In the second process, the model generates all the actions it can perform on these objects. The model then considers the goals of the user and the display representation and information from LTM, and selects one object-action pair. The action is taken and the display is updated.

5.4 NETWORK REPRESENTATION OF KITAJIMA AND POLSON'S LICAI MODEL

The user's goals, the display, the actions, and knowledge stored in LTM are all represented by propositions in the model. Thus, this model is analogous to those models that represent meaning as a network of connections among propositions. This model builds two networks. The first network is composed of a set of propositions that represent the tasks and device goals, the display, knowledge retrieved from LTM by the memory sampling process, and the candidate objects for action. This network selects three candidate objects. The second network has all the elements of the first. In addition, it includes all the possible actions that may be taken with the three candidate objects. This network is responsible for selecting an action object pair. An action object pair requires the following knowledge: "information about the object to be acted on, the function of the action, the physical constraints that must be satisfied in order for the action to be performed, the physical action involved, and the consequences of the action."

5.5 LICAC+ MODEL

As mentioned above, users do not have explicit device goals when first exposed to novel interfaces. They may, however, have expertise in a particular domain or application that underlies their interaction with a computer. Thus, users may have well-formulated task goals. The question then arises, can the action-planning model successfully complete a task using a novel interface—that is, with specific task goals but in the absence of precise device goals? Kitajima and Polson (1995b) determined that this was possible only if the task goals were stated in terms that exactly mimicked the display objects on the screen.

Kitajima and Polson proposed that the formulation of task and device goals is critical to performance. Given a task to accomplish, the model transforms the instructions into "problem schemata" (Kintsch and Greeno, 1985). Schemata are propositional knowledge structures composed of a predicate and admissible arguments. These problem schemata take the instructions and create task and device goals that may then be used in the action-planning model. Very often, a user cannot generate the correct device goal. In these instances, users may be given hints or explicit instructions to perform certain actions on the interface (see Franzke, 1995). In this manner, specific device schema may be learned and stored in LTM for later use. However, the critical problem still remains; task goals must be stated in terms that exactly mimic the display objects on the screen. This limitation is dealt with in the CoLiDeS model.

5.6 COLIDES MODEL

CoLiDeS is an acronym for Comprehension-based Link model of Deliberate Search (Blackmon, Kitajima, Polson, 2005; Blackmon, Kitajima, Polson, 2003; Blackmon, Polson, Kitajima , and Lewis, 2005). CoLideS is similar to the LiCAC+ model in that objects on the screen are first grouped and parsed into distinct regions. The degree of similarity between the items in each region and the user's goal is then determined. CoLiDeS assumes that the user selects an action based on the degree of similarity between their task goal and the display objects that are currently visible. In this regard, the model's behavior is analogous to visual search models. In visual search, similarity between target and distractor items guides attention to select the object most similar to the target for further evaluation. In contrast to visual search models that base similarity on object attributes (color, shape, size, etc.), the ColiDeS model measures similarity based on the semantic content.

Currently, the CoLideS model focuses on Web site navigation. The user's goal is to find specific information, which requires the user to make *n* accurate selections where each selection entails the user "clicking on" a "link" with the mouse. In this regard, the CoLiDes model is similar to models of Web-page navigation that assume that the user follows an "information scent" to successfully acquire the desired information (see Chi, Pirolli, and Pitkow, 2000).

5.7 LATENT SEMANTIC ANALYSIS (LSA)

In CoLiDeS, semantic similarity is determined by Latent Semantic Analysis (LSA) (Landauer, 1998). LSA represents words, sentences, and sentence fragments in a n-dimensional vector space. This representation is based on a machine-learned representation of the user population's understanding of words. The similarity between vectors in this space is a function of the angle between these vectors measured by the cosine of that angle. Since the cosine of angles varies continuously between 1 and -1, the degree of similarity between text strings may be viewed as a correlation, "1" meaning entirely correlated or equivalent "-1" antithetical or negatively correlated, and "0" meaning no correlation or unrelated in meaning. LSA also provides a measure of text familiarity that is correlated to text frequency and embedded knowledge. "Term vector length" measures the degree to which knowledge of a text sting is embedded in the LSA semantic space.

5.8 COGNITIVE WALKTHROUGH FOR THE WEB (CWW) AND ACWW (AUTOMATED CWW)

Using the CoLiDeS model, ACWW was developed to detect and correct Web site navigation design problems. ACWW assumes that the layout of the Web page consists of text headings and text links under those text headings (see Figure 4). ACWW can detect the following errors:

"Weak Sent:" the information the user is searching for is not semantically similar to the links.

"Unfamiliar" text: the title or link is an unfamiliar word that is not a part of the user's everyday vocabulary.

"Competing Headings:" Two or more headings are semantically similar, thus making it increasingly likely that the user will attend to an incorrect region of the Web page.

"Competing Links:" Links are semantically similar; thus, the user may select the incorrect link.

The purpose of ACWW is to flag potential problems so that Web designers may make quick repairs on the most glaring of errors. ACWW is a program that allows the Web designer to enter their Web site for CWW evaluation.

Heading 1 Link A1 Link B1
Heading 2 Link A2 Link B2

Figure 4. Heading link structure used as input to ACWW.

6. APPLICATION OF MODELING METHODS TO HCI

The ACWW Tool was used to analyze components of the HCI for the Knowledge Web (KWeb). The KWeb is a Web-based user decision support tool that provides capability when items of different levels of interest must be shared and discussed in a timely way across members of a workgroup or multiple levels of an organization or command. KWeb allows a user to define Web pages to post items of interest. and then attach links, views, files, and add comments to the items over time. Figure 5 shows an initial state of KWeb prior to its population with "items of interest."

Figure 5. Initial display state of KWeb.

6.1 EXAMPLE KWEB TASK

Assume that a user is given a "posting" task to "Post a green status informative report to the KWeb application." KWeb is capable of uploading a file or entering a Uniform Resource Locator (URL) as a detailed report, but for simplicity, it will be assumed that the report is completely contained within the description. Single clicking on the "plus-sign image" to the right of the word "Reports"(see Figure 5) causes KWeb to open a "Create Reports Item" dialog box (see Figure 6) that contains text boxes for the title of the report and high-level description of the report; also included is a pull-down-list of possible states of the report. The asterisks next to "Title," "Description," and "State" indicate that these fields are required information. Among other objects that also appear in the "Create Reports Item" dialog box are two checkbox fields named "Keyword" and "Options." These fields are not required information. Also found on the dialog box are three more icon objects that allow the user to save/post the report (floppy disk icon), save the report to the clipboard (clipboard icon), and cancel the post report item action (X image icon). The display state changes and the corresponding functional text can be viewed only when the mouse is hovering over any of these three icons.

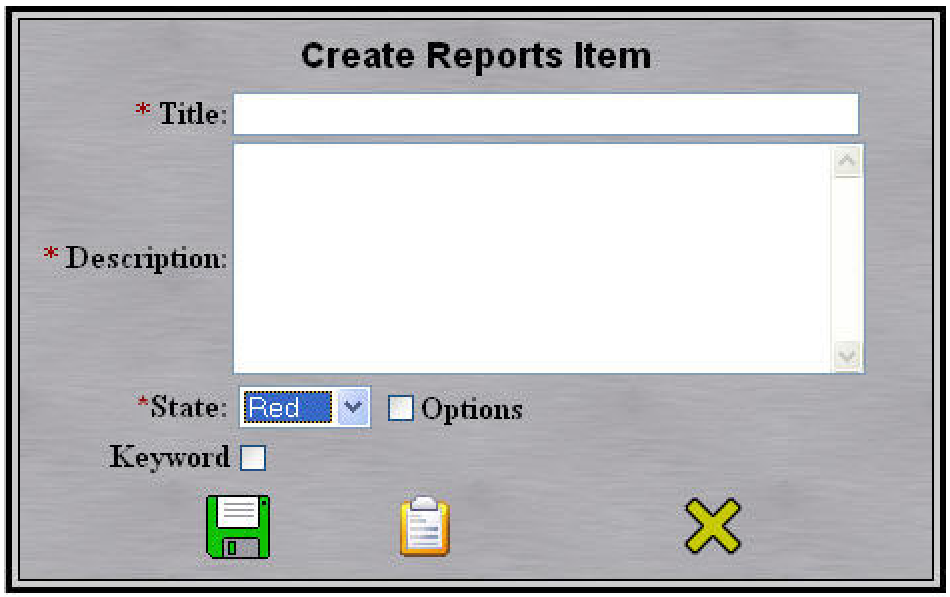


Figure 6. Create Reports Item dialog box.

The posting task can be decomposed into subtasks, where the first subtask is to edit the title of the posted report. This task is accomplished by moving the mouse over the "Title" textbox, single-clicking the mouse, then typing the title. The second subtask is accomplished in a similar way by typing the report into the "Description" textbox. The final subtask is to select the "green" list-item from the "State" pull-down list.

After completing the subtasks, the user must hover the mouse over the floppy-disk image that will popup the text "Save" and single-click to conclude the task. The task of posting a report may be described in terms of user goals sub-goals, display states, and device goals. This analysis is given in Table 1.

The task to post a green status informative report to the KWeb application, and the KWeb interface as depicted in Figures 5 and 6 were evaluated using the ACWW tool. The KWeb interface does not have a heading/link structure. To use the tool, the KWeb interface was adapted to this structure. For example, in the first two trials labeled "Known Hierarchy" (Tables 2 and 3), the words "Home," "Archive," "View," "Status," "Goals," "Reports," and "Insights" were all treated as headings, and the "+'s" were treated as links represented by the word "add." Thus, the "+" sign to the right of the word "Reports" was the "link" under the "heading" report. The semantic similarity of the "Post task..." statement to the "heading" words on the KWeb was evaluated in this manner. The analysis is given in Table 2.

Table 1. Device and task goals for the posting of KWeb task.

Step No.	**Task (TG) and Device goals (DG)**			**Correct Action**
	TG-1	To post an "Informative/No Action Required" (green status) report.		
1		DG-1	See Mouse –over– "add item" associated with the Reports plus-sign image (note there is a caveat for all the pop-up text, if the operator is familiar with the interface they may not need to see the text. So for example the operator can just click on the "+" sign without ever having to see the "add item" text.)	*Move mouse curser to* **+***symbol associated with the Reports.*
2				*Hold mouse in position long enough to reveal the pop up text* ***add item.***
3		DG-2	See the Create Reports Item" dialog box (which obscures part of the original display including the mouse-over text).	*Click on* **+***symbol button.*
	TG-2	Enter title of the report (subgoal).		
4		DG-3	See I-bar cursor inside "Title" text box.	*Move cursor inside Title text box.*
5				*Click on cursor.*
6		DG-4	See correct title in Title text box.	*Type title (using keyboard – note this means the operator takes hand off mouse and places them on keyboard to type.*
	TG-3	Enter report (subgoal).		
7		DG-5	See I-bar cursor inside the "Description" text box.a. Move cursor to inside the description text box. This means that the user must move hands from keyboard back to mouse.	*Move cursor to inside the description text box. This means that the user must move hands from keyboard back to mouse.*
8				*Click on mouse (cursor inside description text box).*
9		DG-6	See text in Description text box.	*Type text – this assumes that the user must move hand from mouse back onto the keyboard in order to enter text.*
10		DG-7	See "green" inside State selection box.	*Move cursor to selection button. This means that the user must move hands from keyboard onto the mouse.*
11				*Click to reveal the pull down selections associated with State.*
12				*Select the green state by moving cursor to the* ***green*** *selection.*
13				*Click mouse.*
14		DG-8	See the mouse over save text associated with the floppy disk icon.	*Move the mouse over the* ***floppy disk*** *icon.*
15		DG-9	Make "create dialog" box disappear revealing the KWeb display that was hidden.	*Click on the save icon.*

Table 2. ACWW analysis output for KWeb page with goal using the word "post." The relationship between the heads and links is given.

Goal: Post a green status informative report to KWeb application.

Predicted Mean Clicks = 2.292 + 1.757 (Link is unfamiliar: false) + 1.516 (Link has a weak-scent: true) +
0.655 * 0 (Number of competing links nested under competing headings) +
0.0 * 0(Number of competing links nested under the correct headings) +
0.0 * 0(Number of competing headings) = 3.808

Heading Frequency: 50.0 Heading Cosine: 0.5 Link Frequency: 50.0 Link Cosine: 0.5 Space: tasaALL Webpage: KWeb

Original Label	*Text*	*Cosine*	*Term Vector*	*Heading Or Link*	*Specific Heading*	*Correct*	*Weak-Scent Correct Link*	*Unfamiliar Correct Link*	*Competing Link under Competing Heading*	*Competing Link under Correct Heading*	*Competing Heading*
Status	status status socioeconomic add add	0.21	0.7	Heading	status status						
Reports	reports reports report reporting clerks accountants accounting financial analyzing reported accountant add add	0.2	0.93	Heading	reports report	X					
Home	home home edit text keyboard displayed typewriter add add	0.06	1.34	Heading	home home edit						
Archive	archive off off on on	-0.01	Can't find	Heading	archive off o						
Goals	goals goals goal objectives achieve outcomes planning add add	-0.02	1	Heading	goals goals go						
View	view view views projection sectional perspective projected gmap other other	-0.04	1.2	Heading	view view view						
Insights	insights add add	-0.05	0.19	Heading	insights add						
On	on on	0.17	0.22	Link	Archive						
Edit	edit text keyboard displayed typewriter	0.09	0.17	Link	Home						
other	other other	0.01	0.5	Link	View						
gmap	gmap	0	Can't find	Link	View						
Off	off off	-0.02	0.95	Link	Archive						
Add	add add	-0.05	0.69	Link	Home						
Add	add add	-0.05	0.69	Link	Status						
Add	add add	-0.05	0.69	Link	Goals						
Add	add add	-0.05	0.69	Link	Insights						
Add	add add	-0.05	0.69	Link	Reports	X	X				

Table 3. ACWW analysis output for KWeb page with goal using the word “add” (second run). The relationship between the headings and links is given.

Goal: Add a green status informative report to KWeb application.

Predicted Mean Clicks = 2.292 + 1.757 (Link is unfamiliar: false) + 1.516 (Link has a weak-scent: false) +
0.655 * 3 (Number of competing links nested under competing headings) +
0.0 * 0(Number of competing links nested under the correct headings) +
0.0 * 3(Number of competing headings) = 4.257

Heading Frequency: 50.0 Heading Cosine: 0.5 Link Frequency: 50.0 Link Cosine: 0.5 Space: tasaALL Webpage: KWeb

Original Label	*Text*	*Cosine*	*Term Vector*	*Heading Or Link*	*Specific Heading*	*Correct*	*Weak-Scent Correct Link*	*Unfamiliar Correct Link*	*Competing Link under Competing Heading*	*Competing Link under Correct Heading*	*Competing Heading*
Status	status status socioeconomic add add	0.37	0.7	Heading	status status						x
Reports	reports reports report reporting clerks accountants accounting financial analyzing reported accountant add add	0.25	0.93	Heading	reports report	x					
Insights	insights add add	0.2	0.19	Heading	insights add						x
Home	home home edit text keyboard displayed typewriter add add	0.12	1.34	Heading	home home edit						x
Goals	goals goals goal objectives achieve outcomes planning add add	0.07	1	Heading	goals goals go						
Archive	archive off off on on	-0.01	Can't find	Heading	archive off o						
View	view view views projection sectional perspective projected gmap other other	-0.02	1.2	Heading	view view view						
Add	add add	0.21	0.69	Link	Insights				x		
Add	add add	0.21	0.69	Link	Home				x		
Add	add add	0.21	0.69	Link	Status				x		
Add	add add	0.21	0.69	Link	Goals						
Add	add add	0.21	0.69	Link	Reports	x					
On	on on	0.15	0.22	Link	Archive						
Edit	edit text keyboard displayed typewriter	0.08	0.17	Link	Home						
other	other other	0.01	0.5	Link	View						
gmap	gmap	0	Can't find	Link	View						
Off	off off	-0.02	0.95	Link	Archive						

To determine what headings the user would attend to and what link the user would select, ACWW considers both the term vector value and cosine of the headings and links. Term vector refers to the degree of familiarity of a string of word. This familiarity may be based on several factors—the user's reading level, their occupation, etc. If the term vector value exceeds some minimum threshold (example 0.5), then the word string is kept for further review. After eliminating the unfamiliar word strings, the model ranks the headings and links according to descending cosine value. For example, using a term vector threshold of 0.5, ACWW would rank the heading 'Status" as the most likely heading to be attended to, since the cosine score of "Status" at 0.21, is slightly higher than "Reports" at 0.2. Note that if the term vector threshold is set to a value greater than 0.7, "Status" would not be considered. The cosine value of status is higher than report since the former matches the heading "Status" perfectly whereas the plural "Reports" is used as a heading instead of the singular "report" used in the goal statement. In either case, the " +" sign link under the "Reports" heading is flagged as having a "weak-sent" since the word "add" was not considered to be semantically similar to the stated goal (cosine = -0.50). This test also demonstrates that LSA did not organize the goal statement in a hierarchical manner, since classifying the report with a "green status" is clearly a sub-goal that is buried in the overall goal structure (see Table 1, DG 7) of posting a report.

In our second run (see Table 3), we kept the heading and link structure the same, but changed the word "Post" to "Add" in the Goal statement: "Add a green status informative report to the KWeb application." In this case, the ACWW did not flag the link "Add" as having weak scent since the cosine of add has now increased from -0.05 to 0.21. Unfortunately, all the "Add" links are now flagged as competing links, and headings that have "Add" as a link are now flagged as competing headings: "Status," "Insights," and "Home." Thus, the LSA is very sensitive to the choice of words used in the goal statement. Evidently "Post" was not considered related to "Add."

In our third run (see Table 4), we assumed that the user did not know the hierarchy between headings and links. Perceptually, we believe that this is a valid assumption because there is very little information that allows one to group the "+" signs with any of the words on the screen. Likewise, there is little or no information to suggest that the plus sign icons "+" are selectable items, whereas the words are not selectable. Thus, the words "Home," "Archive," "View," "Status," "Goals," "Reports," and "Insights" were all treated as headings and links, as well as the plus sign icons, "+". In Table 4 we see that the pattern of results is quite different than those given in Tables 2 or 3. Now all the "+" signs, "Add," are considered competing headings and links as well as the text words "Reports" and "Status." The latter words have higher cosine values than the plus sign icons but they are not selectable. Thus, the user would click on a word, "Report" or "Status," only to find that nothing happens.

The third run demonstrates the importance of perceptual grouping and parsing in determining errors in interface design. The plus signs are not grouped as links subsumed under the text headings. As information is posted to the KWeb, the interface commits a glaring error that involves grouping, which is illustrated in Figure 7. Grouping by proximity would lead one to believe that the exclamation point "!" is associated with the "insight1." The "!" represents a comment that has been made about "report2" and is not at all associated with an "insight1!" This is a striking example of an interface error that violates the perceptual principle of grouping by proximity that should be flagged and brought to the designer's attention.

Table 4. ACWW analysis output for KWeb page with goal using the word “Add” (third run). The relationship between some of the headings and links is not known.

Goal: Add a green status informative report to KWeb application.

Predicted Mean Clicks = 2.292 + 1.757 (Link is unfamiliar: false) + 1.516 (Link has a weak-scent: false) +
0.655 * 6 (Number of competing links nested under competing headings) + 0.0 * 0(Number of competing links nested under the correct headings) +
0.0 * 6(Number of competing headings) = 6.222

Heading Frequency: 50.0 Heading Cosine: 0.5 Link Frequency: 50.0 Link Cosine: 0.5 Space: tasaALL Webpage: KWeb

Original Label	*Text*	*Cosine*	*Term Vector*	*Heading Or Link*	*Specific Heading*	*Correct*	*Weak-Scent Correct Link*	*Unfamiliar Correct Link*	*Competing Link under Competing Heading*	*Competing Link under Correct Heading*	*Competing Heading*
Status	status status socioeconomic status status socioeconomic	0.32	0.7	Heading	status status						x
Reports	reports reports report reporting clerks accountants accounting financial analyzing reported accountant reports reports report reporting clerks accountants accounting financial analyzing reported accountant	0.23	0.93	Heading	reports report						x
Add	add add add add	0.21	0.69	Heading	add add add ad	x					
Add	add add add add	0.21	0.69	Heading	add add add ad						x
Add	add add add add	0.21	0.69	Heading	add add add ad						x
Add	add add add add	0.21	0.69	Heading	add add add ad						x
Add	add add add add	0.21	0.69	Heading	add add add ad						x
Edit	edit text keyboard displayed typewriter edit text keyboard displayed typewriter	0.08	0.17	Heading	edit text keyb						
Goals	goals goals goal objectives achieve outcomes planning goals goals goal objectives achieve outcomes planning	0.02	1	Heading	goals goals go						
Insights	insights insights	0	0.19	Heading	insights insi						
Home	home home home home	-0.01	1.34	Heading	home home home						
Archive	archive off off on on	-0.01	Can't fi	Heading	archive off o						
View	view view views projection sectional perspective projected gmap other other	-0.02	1.2	Heading	view view view						
Status	status status socioeconomic	0.32	0.7	Link	Status				x		
Reports	reports reports report reporting clerks accountants accounting financial analyzing reported accountant	0.23	0.93	Link	Reports				x		
Add	add add	0.21	0.69	Link	Add	x					
Add	add add	0.21	0.69	Link	Add				x		
Add	add add	0.21	0.69	Link	Add				x		
Add	add add	0.21	0.69	Link	Add				x		
Add	add add	0.21	0.69	Link	Add				x		
On	on on	0.15	0.22	Link	Archive						
Edit	edit text keyboard displayed typewriter	0.08	0.17	Link	Edit						
Goals	goals goals goal objectives achieve outcomes planning	0.02	1	Link	Goals						
other	other other	0.01	0.5	Link	View						
gmap	gmap	0	Can't fi	Link	View						
Insights	insights	0	0.19	Link	Insights						
Home	home home	-0.01	1.34	Link	Home						
Off	off off	-0.02	0.95	Link	Archive						

7. ACWW APPLICATION CONCLUSIONS AND RECOMMENDATIONS

After completing the study described in this report, the SSC Pacific design team makes the following conclusions and recommendations for application of the ACWW:

a. The AWCC model obviously needs to be extended so that interfaces and Web pages with layouts that differ from the heading/link structure may serve as input to the model.
b. Various icons, images, and widgets such as text boxes and radio buttons must be entered into the model.
c. The model must be capable of representing the semantic information associated with icons, images, and widgets.
d. The model must perform a perceptual parsing and grouping of information independent of the designers intended structure. The designer must also be able to input their intended structure/grouping of items on the interface. In this manner, the designer’s structure may be evaluated for errors in parsing and grouping (see Figure 7).
e. In addition to grouping errors, the model should be capable of flagging errors involving semantic consistency within the interface. For example, in the KWeb, the plus sign icon is used to add a report, but when the report is to be uploaded, the icon switches to a paper clip. Why the change in the icon? Likewise, in Figure 6 the word “description” appears, but if the report is typed in the text box labeled “description” it is not a description of the report but the report itself. (If the report is an uploaded attached file then the text box must contain a brief description of the report.) So, labeling the text box “description” is ambiguous and inconsistent.
f. These last two analyses concerning grouping and consistency are not limited to the operator performing a specific task. They address generic problems that may arise in web page and interface design. A model should flag generic interface errors, that is, errors that are not task-specific.
g. Lastly, we believe that the strategy to search an interface based on similarity is modulated by expectation. For example, if an initial selection, based on similarity, turns out to be the wrong move, the user’s sense of expectation has been violated, which may lead the user to explore the interface in a manner that is quite independent of similarity. Likewise, the user may decide to change how they are calibrating similarity. Similarity metrics should be adaptive.

Figure 7. Examples of grouping error in KWeb design.

8. REFERENCES

Alterman, R. 1999. “The closed room.” *Mind, Culture and Activity*, vol. 6, no. 2, pp. 117–142.

Blackmon, M. H., M. Kitajima, and P. G. Polson. 2005. “Tool for Accurately Predicting Website Navigation Problems, Non-problems, Problem Severity, and Effectiveness of Repairs.” *Proceedings of the SIGCHI Conference on Human Factors in Computing System*s (pp. 31–40), 2–7 April, Portland, OR, ACM Press.

Blackmon, M. H., M. Kitajima, and P. G. Polson. 2003. “Repairing Usability Problems Identified by the Cognitive Walkthrough for the Web.” *Proceedings of the SIGCHI Conference on Human Factors in Computing System*s (pp. 497–504), 5–10 April, Ft. Lauderdale, FL. ACM Press.

Blackmon, M. H., P. G. Polson, M. Kitajima, and C. Lewis. 2002. “Cognitive Walkthrough for the Web.” *Proceedings of the SIGCHI Conference on Human Factors in Computing System*s (pp. 463–470), 20–25 April, Minneapolis, MN. ACM Press.

Card, S., T. Moran, and A. Newell. 1983. *The Psychology of Human-Computer Interaction*. Lawrence Elbaum Associates, Hillsdale, NJ.

Chi, E. H., P. Pirolli, and J. Pitkow. 2000. “A System for Analyzing and Predicting Information Scent, Usage and Usability of a Web Site.” *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (pp. 161–168), 1–6 April, The Hague, The Netherlands. ACM Press.

Constantine, L. L. 1998. “Prototyping from the User’s Viewpoint.” In *The Unified Process Elaboration Phase: Best Practices in Implementing the UP*, pp. 115–122. CMP Books, Lawrence, KS.

Constantine, L. L. and L. A. D. Lockwood. 1999. “Software for Use: A Practical Guide to the Models and Methods of Usage Centered Design.” Addison-Wesley, Boston, MA.

Franzke, M. 1995. “Turning Research into Practice: Characteristics of Display-Based Interaction.” *Proceedings of Association for Computing Machinery (ACM) CHI ’95* (pp. 421–428), 7–11 May 1995, Denver, CO. ACM Press.

Fowler, M. 2005. “The New Methodology.” URL: http://www.martinfowler.com/articles/newMethodology.html

Hutchins, E. L., J. D. Holland, and D. A. Norman. 1986. “Direct Manipulation Interfaces.” In *User-Centered System Design*, pp. 87–124, D. A Norman and S. W. Draper, Eds. Lawrence Erlbaum Associates, Hillsdale, NJ.

Kieras, D. E. 1997. “A Guide to GOMS Model Usability Evaluation using NGOMSL.” In *Handbook of Human–Computer Interaction*, 733–766, M. Helander, T. Landauer, & P. Prabhu, Eds. 2nd ed. Elsevier Science Inc., New York, NY.

Kieras, D.E. In press. “Model Based Evaluation.” *Handbook for Human-Computer Interaction*, J. Jacko and A. Sears, Eds. Lawrence Erlbaum Associates, Hillsdale, NJ.

Kintsch, W. 1988. “The Role of Knowledge in Discourse: A Construction-integration Model,” *Psychological Review*, pp. 163–182.

Kintsch, W. and J. G. Greeno. 1985. “Understanding and Solving Word Arithmetic Problems,” *Psychological Review*, vol. 92, pp. 109–129.

Kitajima, M. and P. G. Polson. 1995a. “A Comprehension-based Model of Correct Performance and Errors in Skilled, Display-based Human–Computer Interaction,” *International Journal of Human-Computer Systems*, vol. 43, pp. 65–99.

Kitajima, M. and P. G. Polson. 1995b. “LICAI+: A Comprehensive-based Model of Correct Performance in Skilled Display-based Human–Computer Interaction, *International Journal of Human-Computer Studies*, vol. 43, no. 1, pp. 65–99.

Kitajima, M. and P. G. Polson. 1997. “LICAI+: A Comprehensive-based Model of Learning for Display-based Human–Computer Interaction.” *Extended Abstracts on Human Factors in Computing Systems* (pp. 333–334), 22–27 March, Atlanta, GA, ACM Press.

Landauer, T. K., P. W. Foltz, and D. Laham. 1998. “An Introduction to Latent Semantic Analysis,” *Discourse Processes*, vol. 25, pp. 259–284.

Mannes, S. M. and W. Kintsch. 1991. “Routine Computing Tasks: Planning as Understanding,” *Cognitive Science*, vol. 15, pp. 305–342.

Nielsen, J. and R. Molich. 1990. “Heuristic Evaluation of User Interfaces,” *Proceedings of the ACM CHI’90 Conference* (pp. 249–256), 1–5 April, Seattle, WA.

Nielsen, J. 1993. *Usability Engineering*. Academic Press, Inc., Boston, MA.

Payne, P. G., H. R. Squibb, and A. Howes. 1990. “The Nature of Device Models: The Yoked State Hypothesis and Some Experiments with Text Editors,” *Human-Computer Interaction*, vol. 5, pp. 415–444.

Santoro T. and D. E. Kieras. In preparation. “Verification and Validation of GOMS Models for Team Communications and Collaboration.”

Shneiderman, B. 1982. “The Future of Interactive Systems and the Emergence of Direct Manipulation,” *Behavior and Information Technology*, vol. 1, pp. 237–256.

Wolfe, J. M., K. R. Cave, and S. L. Franzel. 1989. “Guided Search: An Alternative to the Feature Integration Model of Visual Search,” *Journal of Experimental Psychology: Human Perception and Performance*, vol. 15, no. 3, pp. 419–433

9. BIBLIOGRAPHY

Chi, E. H., P. Pirolli, K. Chen, and J. Pitkow. 2001.“Using Information Scent to Model User Information Needs and Actions and the Web. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (pp. 490–497), 31 March–5 April, Seattle, WA. ACM Press.

Chi, E. H., A. Rosien, G. Suprattanasiri, A. Williams, C. Royer, C. Chow, E. Robles, B. Dalal, J. Chen, and S. Cousins. 2003. “The Bloodhound Project: Automating Discovery of Web Usability Issues using InfoScent Simulator.” *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (pp. 505–512), Ft. Lauderdale, FL. ACM Press.

Gibson, J. J. 1979. *The Ecological Approach to Visual Perception.* Houghton Mifflin Company, Boston, MA.

John, B. E. and D. E. Kieras. 1996. “The GOMS Family of User Interface Analysis Techniques: Comparison and Contrast,” *ACM Transactions on Computer-Human Interaction,* vol. 3, pp. 320–351.

Landauer, T. K. and S. T. Dumais. 1997. “A Solution to Plato’s Problem: Latent Semantic Analysis Theory of Acquisition Induction and Representation of Knowledge,” *Psychological Review,* vol. 104, no. 2, pp. 211–240.

Landauer, T. K. 1998. “Learning and Representing Verbal Meaning: Latent Semantic Analysis Theory,” *Current Directions in Psychological Science*, vol. 7, pp. 161–164.

Larkin, J. H. 1989. “Display-based Problem Solving.” In *Complex Information Processing: the Impact of Herbert A. Simon*, pp. 319–342, D. Klahr and K. Kotovsky, Eds. Lawrence Erlbaum Associates, Hillsdale, NJ.

Larkin, J. H. and H. A. Simon. 1987. “Why a Diagram Is (Sometimes) Worth 10,000 Words.” *Cognitive Science,* vol. 11, pp. 65–100.

Newell, A. and H. Simon. 1972. *Human Problem Solving*. Prentice-Hall, Englewood Cliffs, NJ.

Pirolli, P. (in press). “The Use of Proximal Information Scent to Forage for Distal Content on the World Wide Web.” In *Working with Technology in Mind: Brunswikian Resources for Cognitive Science and Engineering,* A Kirlik, Ed. Oxford University Press, Oxford, UK.

Pirolli, P. and S. Card. 1999. “Information Foraging.” Xerox Palo Alto Research Center Inc. User Interface Research (UIR) Technical Report, p. 84, Palo Alto, CA.

Pirolli, P. 1997. “Computational Models of Information Scent-Following in a Very Large Browsable Text Collection.” *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 3–10, 22–27 March, Atlanta, GA, ACM Press.

Pirolli, P. and W-T. Fu. 1999. “SNIF-ACT: A Model of Information Foraging on the World Wide Web.” *9th International Conference on User Modeling. Lecture Notes in Artificial Intelligence,* vol. 2702, pp. 45–54.

Polson, P. G. and C. H. Lewis. 1990. “Theory-based Design for Easily Learned Interfaces,” *Human–Computer Interaction,* vol. 5, pp. 191–220.

10. ACRONYMS

ACWW	Automated CWW
ADW	Air Defense Warfare
CoLiDeS	Comprehension-based Link model of Deliberate Search
CPM-GOMS	Cognitive Perceptual Motor GOMS
CWW	Cognitive Walkthrough for the Web
GOMS	Goals, Operators, Methods, and Selection rules
HCI	Human–Computer Interface
HSOC	Hawaii Security Operations Center
JICPAC	Joint Intelligence Command Pacific
KRSOC	Kunia Regional SIGINT Operations Center
KWeb	Knowledge WEB
LACS	Land Attack Combat Systems
LICAI	Linked model of Comprehension-based Action Planning and Instruction-taking
LSA	Latent Semantic Analysis
LTM	Long-Term Memory
MMWS	Multi-Modal Watchstation
PACOM	Pacific Command Hawaii
SSC Pacific	SPAWAR Systems Center Pacific
UCD	User Centered Design
URL	Uniform Resource Locator

REPORT DOCUMENTATION PAGE	*Form Approved* *OMB No. 0704-01-0188*

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
December 2008	Final	

4. TITLE AND SUBTITLE

RAPID, AGILE MODELING SUPPORT FOR HUMAN–COMPUTER INTERFACE CONCEPTUAL DESIGN

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

6. AUTHORS

J. DiVita
R. Morris
G. Osga

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

SSC Pacific
San Diego, CA 92152–5001

8. PERFORMING ORGANIZATION REPORT NUMBER

TR 1976

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Office of Naval Research, Code 31
800 North Quincy Street
Arlington, VA 22217–5660

10. SPONSOR/MONITOR'S ACRONYM(S)

ONR

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

This is a work of the United States Government and therefore is not copyrighted. This work may be copied and disseminated without restriction. Many SSC Pacific public release documents are available in electronic format at http://www.spawar.navy.mil/sti/publications/pubs/index.html

14. ABSTRACT

SPAWAR Systems Center Pacific (SSC Pacific) has developed prototype combat systems. The method of evaluating design alternatives for these prototypes is through an iterative cycle of design, build, and test. This approach has several drawbacks. For example, design information that is critical to support cognitive and perceptual processes in the task domain is not explicitly captured by the design process, but rather is implicitly embodied in the final design. To overcome the problems of this approach, SSC Pacific has explored the feasibility of a model-based approach to system design. This report reviews several models that may be suited to predict early design usability testing.

15. SUBJECT TERMS

Mission Area: Human Factors Engineering

interface design	usability testing	rapid prototyping	agile design process	construction-integration theory	text comprehension
latent semantic	cognitive analysis	conceptual design	keystroke-level model	walkthrough for the web	

16. SECURITY CLASSIFICATION OF: a. REPORT	b. ABSTRACT	c. THIS PAGE	17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
					J. Divita
					19B. TELEPHONE NUMBER *(Include area code)*
U	U	U	UU	39	(619) 553–6461

INITIAL DISTRIBUTION

84300	Library	(2)
85300	S. Baxley	(1)
85300	Archive/Stock	(1)
53621	J. DiVita	(10)

Defense Technical Information Center
Fort Belvoir, VA 22060–6218 (1)

SSC San Diego Liaison Office
C/O PEO-SCS
Arlington, VA 22202–4804 (1)

Center for Naval Analyses
Alexandria, VA 22311–1850 (1)

Government-Industry Data Exchange
Program Operations Center
Corona, CA 91718–8000 (1)

Approved for public release; distribution is unlimited.

SSC Pacific
San Diego, CA 92152-5001